

# 大件摆轮分拣机以太网通信协议

版本：v2.0.3

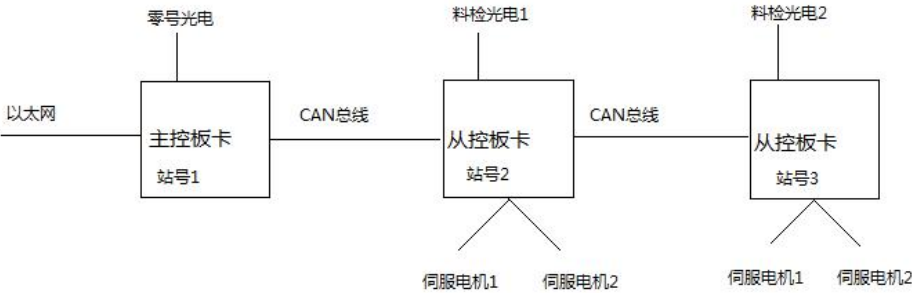
## 目录

一、 通讯简介.....	3
1.1 板卡连接示意图.....	3
1.2 通讯协议格式.....	3
1.3 通讯规则.....	3
二、 通讯命令详解.....	4
2.1 初始化信息(下料口分配表).....	4
2.2 初始化信息(下料口到达光电时间表).....	4
2.2.1 初始化信息 (摆轮数量与每台摆轮驱动器的数量).....	5
2.2.2 初始化最小分拣间隔（小于间隔值不分拣）.....	6
2.3.1 初始化信息(下料口模组的控制参数).....	7
2.3.2 初始化信息(左右分开角度控制).....	8
2.3.3 左右分角度控制参数读取.....	9
2.4 分拣命令.....	10
2.5 分拣结果回应.....	11
2.5.1 包裹间距过小的检测结果输出.....	12
2.6.1 报警信息 1,摆轮报警.....	12
2.6.2 报警信息 2，外部电机（皮带机）报警.....	13
2.6.3 报警信息 3,摆轮电滚筒报警（无 O 带摆轮驱动器通信）.....	14
2.6.4 报警信息 4,摆轮报警(4 个伺服电机后 2 个报警).....	14
2.7 发送给上位机的光电信号.....	15
2.7.1 发送给上位机包裹的长度信息（遮挡时间）.....	15
2.8 板卡心跳命令.....	16
2.9 测试下料光电触发时间命令-1.....	16
2.10 测试下料光电触发时间命令-2.....	17
2.11 读取板卡参数命令.....	17
2.12 写入板卡参数命令.....	18
2.13 初始化信息(靠边区域时间参数).....	19
2.14 下料口开关信号.....	20
2.15 调速命令.....	21
2.16 双驱动器速度查询.....	22
2.17 线体启停状态查询.....	22
2.18 线体启停控制.....	23
2.19 摆轮急停状态查询.....	24
2.20 摆轮控制模式查询.....	24
2.21 包裹补格口数据上传上位机.....	25
2.22 版本号信息上传.....	26
2.23 光电滤波时间设置.....	26
2.24 超时信息追加输出.....	27

2.26 备用光电分拣结果上传（默认使用前一个光电做备用） .....	27
2.27 从板未执行主板发送动作.....	28
2.28.1 分拣时是否加速功能初始化.....	28
2.28.2 分拣时是否加速功能配置读取.....	29
2.30 备用光电分拣功能初始化.....	30
2.31 驱动器站号组别设置.....	31
2.32 驱动器站号组别读取.....	31
2.33 上位机掉线强制异常口设置.....	32
2.34 初始化信息丢失后的心跳.....	33
2.35 使用过程中初始化格口信息.....	33
2.36 是否启用急停按钮.....	34
3.0 不允许分拣命令.....	35
3.1 摆轮单机分拣结果回应.....	35
3.2 供包台是否允许供件信息上传.....	36
3.3 存储配置文件.....	37
3.4 读取配置文件.....	38
*关于重发机制的说明.....	39

# 一、通讯简介

## 1.1 板卡连接示意图



## 1.2 通讯协议格式

数据帧格式：

帧头	帧序号	帧长度	帧校验	帧命令	数据区
----	-----	-----	-----	-----	-----

- ▶ 帧 头：2 个字节，区分每一帧数据，帧头约定为 “0xAAAA”。
- ▶ 帧序号：4 个字节，发送方每一帧自动加 1，应答方应答时使用接收到发送方使用的帧序号。
- ▶ 帧长度：2 个字节，表示整个帧的数据长度。
- ▶ 帧校验：1 个字节，对帧数据中的帧命令和数据区的数据进行异或计算。
- ▶ 帧命令：2 个字节，帧的命令类型代码，表示该帧数据的功能，发送命令最高位为 0，应答方使用的应答命令为接收到的命令最高位或 1。
- ▶ 数据区：每条命令的具体数据，详见命令说明。

注：通信协议帧中的数据全部为小端格式。

## 1.3 通讯规则

上位机系统采用 UDP 方式和下位板卡进行通讯。上位机使用固定 IP：192.168.2.100 port：9000，板卡使用固定 IP：192.168.2.110 port：9000。

## 二、通讯命令详解

### 2.1 初始化信息(下料口分配表)

此协议：自动供包摆轮共用

(1) 发送时间  
上位软件打开后主动发送.

(2) 发送帧命令：0x1A01

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char unloadPortTableNum;
    sUnloadPortPara unloadPortParaTable[unloadPortTableNum];
} sPC2Board_UnloadPortPara;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	unloadPortTableNum	下料口个数
sUnloadPortPara	unloadPortParaTable[unloadPortTableNum]	具体的每个下料口的参数信息

每个下料口参数信息格式如下：

```
typedef struct{
    unsigned char unload_port_num; //下料口号
    unsigned char broad_num;      //从控站号
    unsigned char unload_port_dir; //下料口方向
} sUnloadPortPara;
```

(4) 应答帧命令：0x9A01

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_UnloadPortPara_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

### 2.2 初始化信息(下料口到达光电时间表)

此协议：自动供包摆轮共用

(1) 发送时间

上位软件打开后主动发送.

(2) 发送帧命令: **0x1A02**

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char unloadPhotoNum;
    unsigned int32 unloadPortTime[unloadPhotoNum];
} sPC2Board_UnloadPortTime;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	unloadPhotoNum	下料口光电个数
unsigned int32	unloadPortTime[unloadPhotoNum]	从零号光电开始到下料口光电的时间 ms

(4) 应答帧命令: **0x9A02**

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_UnloadPortTime_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.2.1 初始化信息 (摆轮数量与每台摆轮驱动器的数量)

此协议: 自动供包摆轮共用

(1) 发送时间

上位软件打开后主动发送.

(2) 发送帧命令: **0x1A05**

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char balance_num;
    unsigned char sSwingingWheelData[SwingingWheel_Count];
} sPC2Board_SwingingWheel_Param;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	SwingingWheel_Count	摆轮数量
sSwingingWheelData	SwingingWheelData [SwingingWheel_Count]	摆轮参数

```
typedef struct {
    unsigned char SwingingWheel_Index;    //摆轮（板卡）序号;
```

```
unsigned char SwingingWheel_Type;
```

```
//摆轮类型（1：单驱铁摆轮（或者单驱 O 带摆轮）、2：双驱铁摆轮（或者双驱 O 带摆轮）、3：多驱铁摆轮（或者多驱 O 带摆轮）、4：GSD195-M33P27 摩擦&包胶摩擦摆轮（多驱）、5：GSD195-M24P27 摩擦&包胶摩擦摆轮（多驱）、6：GSD195-M222P27 摩擦&包胶摩擦摆轮（多驱）、7：GSD120-M2233P85 4K 大包胶摆轮（双驱）、8：GSD120-M2233P85 4K 大包胶摆轮（多驱）9：GSD120-M2222P85 6K 大包胶摆轮、10：GSD90-M2223P59 自动供包小包胶摆轮（双驱）、11：GSD90-M2223P59 自动供包小包胶摆轮（多驱）
```

摆轮类型名称解释：举例（GSD195-M33P27 摩擦&包胶摩擦摆轮（GSD195 代表单元直径，195 一般都是大盘。M33，代表每组伺服电机带动的单元排数，33 此处代表两个电机、每个电机带动三排，P27 代表单元个数）

```
unsigned char Driver_Count;          //每台摆轮的驱动器数量
} sSingingWheelData
```

（4）应答帧命令：0x9A05

（5）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_SwingingWheel_Param_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.2.2 初始化最小分拣间隔（小于间隔值不分拣）

（1）发送时间

上位软件打开后主动发送.（如果上位机没有发送或者发送的数值是 0 则不启用此功能）

（2）发送帧命令：0x1A06

（3）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 minInterval;
} sPC2Board_MinInterval_Param;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int16	minInterval	第一个包裹尾部到下个包裹头部的最小间隔时间

（4）应答帧命令：0x9A06

（5）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_MinInterval_Param_ACK;
```

### 2.3.1 初始化信息(下料口模组的控制参数)

此协议: 自动供包摆轮共用

- (1) 发送时间  
上位软件打开后主动发送. 此参数需要写入 Flash 中。
- (2) 发送帧命令: **0x1A03**
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char broad_num;
    unsigned char module_num;
    unsigned int16 photo_dly;
    unsigned int16 start_dly;
    unsigned int16 motor_follow_t;
} sPC2Board_MotorCtrlPara;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	broad_num	板卡的站号
unsigned char	unloadPhotoNum	一个下料口包含的模组数(电机数)
unsigned int16	photo_dly	光电触发延时时间
unsigned int16	start_dly	电机延时动作时间（伺服电机 1，2 的延时启动参数）
unsigned int16	motor_follow_t	跟随电机延时启动时间（伺服电机 2 在延时启动参数基础上再附加一个延时参数）

- (4) 应答帧命令: **0x9A03**
- (5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_MotorCtrlPara_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

(注意:每一个主从控盒子都需要发送这样一个控制参数,而且可以分别配置)

## 2.3.2 初始化信息(左右分开角度控制)

### (1) 发送时间

上位软件打开后主动发送. 配置文件有此协议时不再发送上面的 0x1A03 协议。此参数需要写入 Flash 中。每个板卡的参数单独发送

### (2) 发送帧命令：0x1A12

### (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char Balance_id          //板卡 id
    unsigned char unloadPhotoNum      //一个下料口包含的模组数(电机数)
    unsigned int16    zero_start_dly          //0 方向启动延时时间
    unsigned int16    zero_motor_follow_t1    //0 方向跟随延迟时间 1
    unsigned int16    zero_motor_follow_t2    //0 方向跟随延迟时间 2
    unsigned int16    zero_motor_follow_t3    //0 方向跟随延迟时间 3
    unsigned int16    one_start_dly           //1 方向启动延时时间
    unsigned int16    zero_motor_follow_t1    //1 方向跟随延迟时间 1
    unsigned int16    zero_motor_follow_t2    //1 方向跟随延迟时间 2
    unsigned int16    zero_motor_follow_t3    //1 方向跟随延迟时间 3
    unsigned int16    zero_pulse_1 //0 方向脉冲 1
    unsigned int16    zero_pulse_2 //0 方向脉冲 2
    unsigned int16    zero_pulse_3 //0 方向脉冲 3
    unsigned int16    zero_pulse_4 //0 方向脉冲 4
    unsigned int16    one_pulse_1  //1 方向脉冲 1
    unsigned int16    one_pulse_2  //1 方向脉冲 2
    unsigned int16    one_pulse_3  //1 方向脉冲 3
    unsigned int16    one_pulse_4  //1 方向脉冲 4
} sPC2Board_MotorCtrlPara;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	broad_num	板卡的站号
unsigned char	unloadPhotoNum	一个下料口包含的模组数(电机数)
unsigned int16	zero_start_dly	0 方向启动延时时间
unsigned int16	zero_motor_follow_t1	0 方向跟随延迟时间 1
unsigned int16	zero_motor_follow_t2	0 方向跟随延迟时间 2
unsigned int16	zero_motor_follow_t3	0 方向跟随延迟时间 3
unsigned int16	one_start_dly	1 方向启动延时时间
unsigned int16	zero_motor_follow_t1	1 方向跟随延迟时间 1
unsigned int16	zero_motor_follow_t2	1 方向跟随延迟时间 2
unsigned int16	zero_motor_follow_t3	1 方向跟随延迟时间 3
unsigned int16	zero_pulse_1	0 方向脉冲 1



unsigned int16	zero_pulse_2	0 方向脉冲 2
unsigned int16	zero_pulse_3	0 方向脉冲 3
unsigned int16	zero_pulse_4	0 方向脉冲 4
unsigned int16	one_pulse_1	1 方向脉冲 1
unsigned int16	one_pulse_2	1 方向脉冲 2
unsigned int16	one_pulse_3	1 方向脉冲 3
unsigned int16	one_pulse_4	1 方向脉冲 4

(4) 应答帧命令: **0x9A12**

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_MotorCtrlPara_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

(注意:每一个主从控盒子都需要发送这样一个控制参数,而且可以分别配置)

### 2.3.3 左右分角度控制参数读取

(1) 发送时间

上位软件打开后主动发送.

(2) 发送帧命令: **0x1A13**

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char Balance_id      //板卡 id
} sPC2Board_MotorCtrlPara_Upload;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	broad_num	板卡的站号

(4) 应答帧命令: **0x9A13**

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char Balance_id      //板卡 id
    unsigned char unloadPhotoNum //一个下料口包含的模组数(电机数)
    unsigned int16 zero_start_dly //0 方向启动延时时间
    unsigned int16 zero_motor_follow_t1 //0 方向跟随延迟时间 1
    unsigned int16 zero_motor_follow_t2 //0 方向跟随延迟时间 2
```

```

unsigned int16    zero_motor_follow_t3  //0 方向跟随延迟时间 3
unsigned int16    one_start_dly        //1 方向启动延时时间
unsigned int16    zero_motor_follow_t1  //1 方向跟随延迟时间 1
unsigned int16    zero_motor_follow_t2  //1 方向跟随延迟时间 2
unsigned int16    zero_motor_follow_t3  //1 方向跟随延迟时间 3
unsigned int16    zero_pulse_1  //0 方向脉冲 1
unsigned int16    zero_pulse_2  //0 方向脉冲 2
unsigned int16    zero_pulse_3  //0 方向脉冲 3
unsigned int16    zero_pulse_4  //0 方向脉冲 4
unsigned int16    one_pulse_1  //1 方向脉冲 1
unsigned int16    one_pulse_2  //1 方向脉冲 2
unsigned int16    one_pulse_3  //1 方向脉冲 3
unsigned int16    one_pulse_4  //1 方向脉冲 4
} sPC2Board_MotorCtrlPara_Upload_ack;

```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	broad_num	板卡的站号
unsigned char	unloadPhotoNum	一个下料口包含的模组数(电机数)
unsigned int16	zero_start_dly	0 方向启动延时时间
unsigned int16	zero_motor_follow_t1	0 方向跟随延迟时间 1
unsigned int16	zero_motor_follow_t2	0 方向跟随延迟时间 2
unsigned int16	zero_motor_follow_t3	0 方向跟随延迟时间 3
unsigned int16	one_start_dly	1 方向启动延时时间
unsigned int16	zero_motor_follow_t1	1 方向跟随延迟时间 1
unsigned int16	zero_motor_follow_t2	1 方向跟随延迟时间 2
unsigned int16	zero_motor_follow_t3	1 方向跟随延迟时间 3
unsigned int16	zero_pulse_1	0 方向脉冲 1
unsigned int16	zero_pulse_2	0 方向脉冲 2
unsigned int16	zero_pulse_3	0 方向脉冲 3
unsigned int16	zero_pulse_4	0 方向脉冲 4
unsigned int16	one_pulse_1	1 方向脉冲 1
unsigned int16	one_pulse_2	1 方向脉冲 2
unsigned int16	one_pulse_3	1 方向脉冲 3
unsigned int16	one_pulse_4	1 方向脉冲 4

## 2.4 分拣命令

### (1) 发送时间

上位机收到下料口光电信号后查找到相应的下料口然后发送分拣命令。

(2) 发送帧命令: 0x1B01

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 msg_id;
    unsigned char unload_port_num;
    unsigned int16 time_delay;
    unsigned int16 photo_trigger_time; (这个长度值 V9 会发送, 其他版本不会发送)
} sPC2Board_unload_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int32	msg_id	下料命令序号
unsigned char	unload_port_num	下料口编号
unsigned int16	time_delay	从相机触发光电开始到收到下料信息经过的时间
unsigned int16	photo_trigger_time	从相机触发光电上升沿到下降沿时间 (这个长度值 V9 会发送, 其他版本不会发送)

(4) 应答帧命令: 0x9B01

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 package_id; (包裹号,老版本板卡的 ACK 没有包裹号, 能发送版本号的板卡会带有包裹号, 需要 3 次最多重发功能配合使用)
} sPC2Board_unload_Data_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.5 分拣结果回应

(1) 发送时间

下位机执行下料动作后回应上位机

(2) 发送帧命令: 0x1B02

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char msg_num;
    sMsgData msg_data[msg_num];
} sPC2Board_unload_Result;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	msg_num	回应的消息个数
sMsgData	msg_data[msg_num]	消息数据部分

消息部分数据格式如下：

```
typedef struct{
    unsigned char cmd_type;    //回应消息的类型(0:正常分拣 1:下料超时 2:下料口关闭
    3:下料失败 4:包裹间距小于设定值 5:从板执行分拣动作超时(摆轮不回证版本不适用))
    unsigned int32 msg_id;    //下料命令序号
} sMsgData
(4) 应答帧命令：0x9B02
(5) 数据格式定义
typedef struct{
    unsigned char cmd[11];
    unsigned int32 package_id; (包裹号,板卡收到 ACK 或者达到最大重发次数才会从队列
    中移除此包裹的分拣结果，需要 3 次最多重发功能配合使用)
} sPC2Board_unload_Result_ACK;
```

## 2.5.1 包裹间距过小的检测结果输出

(1) 发送时间  
板卡判断包裹间距过小后追加发送

(2) 发送帧命令:0x1B04

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 pkg_id;
    unsigned int16 pkg_length;
    unsigned int16 interval_before;
    unsigned int16 interval_after;
} sPC2Board_Pkg_Interval;
```

(4) 应答帧命令：无需应答

## 2.6.1 报警信息 1,摆轮报警

此协议：自动供包摆轮共用

(1) 发送时间

板卡每隔 1 秒钟向上位机发送报警信息。

(2) 发送帧命令：0x1C01

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char error_num;
    sAlarmData error_data[error_num];
} sBoard2PC_Alarm_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	error_num	有报警的控制盒的数量
sAlarmData	error_data[error_num]	每个控制盒的报警数据

报警数据格式如下：

```
typedef struct{
    unsigned char broad_num;    //控制器站号
    unsigned char error_status; //错误状态(bit0:零号光电/料检光电,bit1:伺服电机 1, bit2:
    伺服电机 2,bit3:变频器,bit4:接近开关 1,bit5:接近开关 2,bit6:通信)
} sAlarmData
(4) 应答帧命令：无需应答
```

## 2.6.2 报警信息 2，外部电机（皮带机）报警

(1) 发送时间

板卡每隔 1 秒钟向上位机发送报警信息。

(2) 发送帧命令：0x1801

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char error_num;
    sExMotorData error_data[error_num];
} sBoard2PC_EXMotor_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	error_num	有报警外部电机总个数
sAlarmData	error_data[error_num]	每个外部电机的报警信息

报警数据格式如下：

```
typedef struct{
    unsigned char exmotor_count;    //外部电机序号;
    unsigned char error_status;    //错误状态（0x1：电机报警）
} sEXMotorAlarmData
```

(4) 应答帧命令：无需应答

### 2.6.3 报警信息 3,摆轮电滚筒报警（无 O 带摆轮驱动器通信）

此协议：自动供包摆轮共用

(1) 发送时间

检测到有驱动器报警状态时上传，一秒一次，一次传一个从站的报警信息，无错误状态不上传。

(2) 发送帧命令：0x1401

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char board_ID;
    unsigned char error_num;
    sDriverData error_data[error_num];
} sBoard2PC_Alarm_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	board_ID	有报警的站号
unsigned char	error_num	有报警的电滚筒总数量
sAlarmData	error_data[error_num]	每个电滚筒的报警数据

报警数据格式如下：

```
typedef struct {
    unsigned char driver_count;    //电滚筒序号;
    unsigned char error_status;    //错误状态
0x01: 滚筒电机堵转
0x02: 滚筒电机过流
0x03: 48V 电源过压/欠压
0x04: 滚筒驱动器报警
0x0f: 滚筒驱动器未连接
} sDriverData
```

(4) 应答帧命令：无需应答

### 2.6.4 报警信息 4,摆轮报警(4 个伺服电机后 2 个报警)

此协议：自动供包摆轮共用

(1) 发送时间

板卡每隔 1 秒钟向上位机发送报警信息。

(2) 发送帧命令：0x1C02

### (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char error_num;
    sAlarmData error_data[error_num];
} sBoard2PC_Alarm_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	error_num	有报警的控制盒的数量
sAlarmData	error_data[error_num]	每个控制盒的报警数据

报警数据格式如下：

```
typedef struct{
    unsigned char broad_num;    //控制器站号
    unsigned char error_status; //错误状态(bit0:伺服电机 3, bit1:伺服电机 4 ,bit2:接近开
    关 3,bit3:接近开关 4)
} sAlarmData
```

### (4) 应答帧命令：无需应答

## 2.7 发送给上位机的光电信号

此协议：自动供包摆轮共用

### (1) 发送时间

板卡收到光电开关触发信号时向上位机发送该命令

### (2) 发送帧命令：0x1D01

### (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char photoSwitch_num;
} sBoard2PC_Photoelectric_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	photoSwitch_num	光电开关序号

### (4) 应答帧命令：无需应答

## 2.7.1 发送给上位机包裹的长度信息（遮挡时间）

### (1) 发送时间

板卡通过光电（一般通过 0 号光电）采集到包裹的长度后向上位机发送该命令

### (2) 发送帧命令：0x1D02

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 pkg_length;
} sBoard2PC_PKG_LENGTH;
```

(4) 应答帧命令：无需应答

## 2.8 板卡心跳命令

此协议：自动供包摆轮共用

(1) 发送时间

板卡与上位机连续 5 秒未发生数据交互时发送。

(2) 发送帧命令：0x1150

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sBoard2PC_Heart_Data;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

(4) 应答帧：无需应答

## 2.9 测试下料光电触发时间命令-1

(1) 发送时间

上位机向板卡发送进入测试模式命令

(2) 发送帧命令：0x1E01

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char test_mode;
} sPC2Board_Test_Mode;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	test_mode	0:普通模式 1:测试模式

(4) 应答帧命令：无需应答



## 2.10 测试下料光电触发时间命令-2

- (1) 发送时间  
板卡把每一个到下料口光电的时间发送到上位机
- (2) 发送帧命令：0x1E02
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char photo_index;
    unsigned int32 photo_time;
} sBoard2PC_Test_Mode;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	photo_index	下料口光电的序号(从 1 开始)
unsigned int32	photo_time	包裹到达该下料口的时间 ms

- (4) 应答帧命令：无需应答

## 2.11 读取板卡参数命令

此协议：自动供包摆轮共用

- (1) 发送时间  
仅在板卡 boot 模式中使用, 用于调试。
- (2) 发送帧命令：0x1F10
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    Unsigned char Board_num;    板卡号
} sPC2Board_Read_Para;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

- (4) 应答帧命令：0x9F10
- (5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 motor_find_zero_cnt[4];
    unsigned int16 motor_speed_modulus;
    unsigned int16 motor_rolling_pulse;
    unsigned int16 drive_available;
    unsigned int16 motor_slave_para_backup [3];
    unsigned int16 photo_delay_para_cnt[2];
```

```

unsigned int16 motor_hold_time;
unsigned int16 motor_master_para_backup[3];
signed int16 compensation[4];
} sPC2Board_Read_Para_ACK;

```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int16	motor_find_zero_cnt[4]	从站参数：电机 1-4 寻零脉冲数，电机寻零脉冲默认 1600(铁摆轮)，O 带摆轮需手动设置为 3300。
unsigned int16	motor_speed_modulus	从站参数：伺服电机转动速度系数。默认值 3，范围 3~33
unsigned int16	motor_rolling_pulse	从站参数：摆动脉冲数（电机转动角度）默认值 2000。
unsigned int16	drive_available	驱动器速度
unsigned int16	motor_slave_para_backup[3]	从站备用参数
unsigned int16	photo_delay_para_cnt[2]	主站参数：光电检测允许提前量，默认值 200。光电检测允许延后量，值默认值 450。
unsigned int16	motor_hold_time	主站参数：电机摆出保持时间，默认值 500
unsigned int16	motor_master_para_backup[3]	主站 boot 参数备用预留
signed int16	Compensation[4]	从站机械间隙补偿参数

## 2.12 写入板卡参数命令

此协议：自动供包摆轮共用

(1) 发送时间

仅在板卡 boot 模式中使用, 用于调试。

(2) 发送帧命令：0x1F11

(3) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
    unsigned int16 motor_find_zero_cnt[4];
    unsigned int16 motor_speed_modulus;
    unsigned int16 motor_rolling_pulse;
    unsigned int16 motor_slave_para_backup [4];
    unsigned int16 photo_delay_para_cnt[2];
    unsigned int16 motor_hold_time;
    unsigned int16 motor_master_para_backup[3];
    signed int16 compensation[4];
}

```

```
} sPC2Board_Write_Para;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int16	motor_find_zero_cnt[4]	从站参数：电机 1-4 寻零脉冲数，电机寻零脉冲默认 1600。
unsigned int16	motor_speed_modulus	从站参数：伺服电机转动速度系数。默认值 3，范围 3~33
unsigned int16	motor_rolling_pulse	从站参数：摆动脉冲数（电机转动角度）
unsigned int16	drive_available	驱动器速度
unsigned int16	motor_slave_para_backup[2]	从站备用参数
unsigned int16	motorCtrlPara.start_dly	启动延时时间
unsigned int16	motorCtrlPara.motor_follow_t	跟随时间
unsigned int16	photo_delay_para_cnt[2]	主站参数：光电检测向前允许误差量，光电检测向后允许误差量
unsigned int16	motor_hold_time	主站参数：电机摆出保持时间
unsigned int16	motor_master_para_backup[3]	主站 boot 参数备用预留
signed int16	Compensation[4]	从站机械间隙补偿参数

（4）应答帧命令：0x9F11

（5）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_Write_Para_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.13 初始化信息(靠边区域时间参数)

（1）发送时间

上位软件打开后主动发送。

（2）发送帧命令：0x1A04

（3）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 keepSideDelayTime;
} sPC2Board_KeepSidePara;
```

变量类型	变量名	含义
------	-----	----

unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int32	keepSideDelayTime	相机触发光电到靠边区的时间（ms）

（4）应答帧命令：**0x9A04**

（5）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_KeepSidePara_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.14 下料口开关信号

（1）发送时间

上位软件在需要时主动发送给板卡。

（2）发送帧命令：**0x1B03**

（3）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char port_num;
    sPortClose portCloseData[port_num];
} sPC2Board_PortClose_Cmd;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	port_num	开关的下料口数量
sPortClose	portCloseData[port_num]	下料口开关数据部分

下料口开关部分数据格式如下：

```
typedef struct{
    unsigned char unload_port_num;    //下料口号
    unsigned char unload_port_close;  //下料开关状态(0: 开  1: 关)
} sPortClose
```

（4）应答帧命令：**0x9B03**

（5）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_PortClose_ACK;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.15 调速命令

此协议: 自动供包摆轮共用

(1) 发送时间

上位软件在需要时主动发送给板卡. 仅限于调节双驱动器驱动的电滚筒

(2) 发送帧命令: **0x1901**

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char driverSpeedValue;
} sPC2Board_DriverSpeedValue_cmd
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	driverSpeedValue	驱动器速度值

驱动器速度值部分数据解释:

速度 (M/S)	调速指令 3 字节(速度只是对于现有摆轮 63.5MM 直径参考, 准确的需要根据实际调节)
1.5	18
1.8	1D (默认值)
2.0	20
2.5	28
3	2F
3.5	38

(4) 应答帧命令: **0x9901**

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_DriverSpeedValue_cmd;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

## 2.16 双驱动器速度查询

此协议：自动供包摆轮共用

(1) 发送时间

上位软件在需要时主动发送给板卡. 仅限于获取双驱动器驱动的电滚筒速度. 在驱动器速度改变的时候主动发送给上位机软件.

(2) 发送帧命令：0x1902

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_QueryDriverSpeed_cmd
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

(4) 应答帧命令：0x9902

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    Unsigned char balance_num;    //摆轮数量
    sDriverSpeedData[balance_num] ;
} sPC2Board_DriverSpeedResult_cmd;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	balance_num	从板卡数量
sDriverSpeedData	sDriverSpeedData[balance_num]	驱动器速度信息

驱动器速度信息

```
typedef struct{
    unsigned char balance_index;
    unsigned char driverSpeedValue;
} sDriverSpeedData
```

## 2.17 线体启停状态查询

此协议：自动供包摆轮共用

(1) 发送时间

上位软件在需要时主动发送给板卡. 在启停状态改变的时候主动发送给上位机软件.

(2) 发送帧命令：0x1201

(3) 数据格式定义

```
typedef struct{
```

```

        unsigned char cmd[11];
    } sPC2Board_QueryIsRunning_cmd

```

unsigned char	cmd[11]	详见 1.1 通讯协议格式
---------------	---------	---------------

(4) 应答帧命令: **0x9201**

(5) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
    unsigned char num;
    srunstate IsRunning[num];    //线体启停状态(0: 停止  1: 运行)
} sPC2Board_IsRunning_cmd;

```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	num	摆轮数量
srunstate	IsRunning[num]	摆轮启停状态信息(0: 停止  1: 运行)

```

typedef struct{
    unsigned char balance_index; 板卡号
    unsigned char runstate_Value; 启停状态信息
} srunstate;

```

## 2.18 线体启停控制

此协议: 自动供包摆轮共用

(1) 发送时间:

上位机软件在需要控制线体启停的时候主动发送给板卡.

(2) 发送帧命令: **0x1202**

(3) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
    unsigned char StartOrStop;
} sPC2Board_StartOrStop_cmd

```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	StartOrStop	启停状态(0: 停止  1: 运行)

(4) 应答帧命令: **0x9202**

(5) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
} sPC2Board_StartOrStop_Ack

```

变量类型	变量名	含义
------	-----	----

unsigned char	cmd[11]	详见 1.1 通讯协议格式
---------------	---------	---------------

## 2.19 摆轮急停状态查询

此协议：自动供包摆轮共用

- (1) 发送时间  
上位软件在需要时主动发送给板卡. 在急停状态改变的时候主动发送给上位机软件.
- (2) 发送帧命令：0x1301
- (3) 数据格式定义
- ```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_QueryEStop_cmd
```

|               |         |               |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |
|---------------|---------|---------------|

- (4) 应答帧命令：0x9301
- (5) 数据格式定义
- ```
typedef struct{
    unsigned char cmd[11];
    unsigned char num;
    EStopState EStop[num];    //线体急停状态(0: 正常 1: 急停)
} sPC2Board_EStop_cmd;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	num	摆轮数量
EStopState	EStop[num]	摆轮启停状态信息(0: 正常 1: 急停)

```
typedef struct{
    unsigned char board_index; 板卡号
    unsigned char EStop_Value; 急停状态信息
} s_EStop_State;
```

## 2.20 摆轮控制模式查询

此协议：自动供包摆轮共用

- (1) 发送时间  
上位软件在需要时主动发送给板卡查询. 上电时板卡将各个从板卡的本地远程状态主动发给上位机
- (2) 发送帧命令：0x1501
- (3) 数据格式定义



```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_QueryEStop_cmd
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式

(4) 应答帧命令: **0x9501**

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char num;
    CModelInfo [num];    //摆轮控制模式(0: 本地  1: 远程)
} sBoard2pc_CModelInfo_cmd;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned char	num	板卡数量
s_CModelInfo	CModelInfo[num]	摆轮控制模式信息(0: 本地  1: 远程)

```
typedef struct{
    unsigned char board_index; 板卡号
    unsigned char Elocal_Value; 本地远程状态信息
} s_CModelInfo;
```

## 2.21 包裹补格口数据上传上位机

(1) 发送时间

现场操作员按下补格口按钮后发送对应格口数据给上位机。

(2) 发送帧命令: **0x1D03**

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 SortPortID;
} sPC2Board_CSP_cmd
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int16	SortPortID	按钮代表的格口 ID

(4) 应答帧命令: **无需应答**

## 2.22 版本号信息上传

此协议：自动供包摆轮共用

- (1) 发送时间  
初始化完成后主动上传一次
- (2) 发送帧命令：0x1E03
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 version;
} sPC2Board_ version;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int32	version	版本号（比如 1.2.1.6 则 对应 0x06 0x01 0x02 0x01）

- (4) 应答帧命令：无需应答

## 2.23 光电滤波时间设置

- (1) 发送时间  
上位机需要时发送，如果板卡收到的值不在合理范围内，则采用板卡内部默认值
- (2) 发送帧命令：0x1E04
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 FilteringTime;
} sPC2Board_ FilteringTime;
```

变量类型	变量名	含义
unsigned char	cmd[11]	详见 1.1 通讯协议格式
unsigned int16	FilteringTime	滤波时长

- (4) 应答帧命令：0x9E04
  - (5) 数据格式定义
- ```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_ FilteringTime_Ack;
```

## 2.24 超时信息追加输出

(1) 发送时间

包裹分拣超时后发送

(2) 发送帧命令：0x1B07

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int32 package_id;
    unsigned char board_id;
    unsigned int32 Cumulative_Time;
    unsigned char msg_num;
    sMsgData msg_data[msg_num];

} sPC2Board_OverTimePackageInfo;
```

| 变量类型           | 变量名               | 含义            |
|----------------|-------------------|---------------|
| unsigned char  | cmd[11]           | 详见 1.1 通讯协议格式 |
| unsigned int32 | package_id        | 包裹 ID         |
| unsigned char  | board_id          | 板卡站号          |
| unsigned int32 | Cumulative_Time   | 累计计时          |
| unsigned char  | msg_num           | 消息数据个数        |
| sMsgData       | msg_data[msg_num] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char board_id;           //板卡站号 ID
    unsigned char check_flag;         //校验标志（0：提前，1：延后）
    unsigned int16 RevisionTime;      //修正时间
} sMsgData
```

(4) 应答帧命令：无需应答

## 2.26 备用光电分拣结果上传（默认使用前一个光电做备用）

(1) 发送时间

下位机通过备用光电执行下料动作后回应上位机

(2) 发送帧命令：0x1B08

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char msg_num;
```

```
sMsgData msg_data[msg_num];
} sPC2Board_unload_Result;
```

| 变量类型          | 变量名               | 含义            |
|---------------|-------------------|---------------|
| unsigned char | cmd[11]           | 详见 1.1 通讯协议格式 |
| unsigned char | msg_num           | 回应的消息个数       |
| sMsgData      | msg_data[msg_num] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char cmd_type;    //回应消息的类型(0:正常分拣 1:下料超时 2:下料口关闭
    3:下料失败 4:包裹间距小于设定值 5:从板执行分拣动作超时(摆轮不回证版本不适用))
    unsigned int32 msg_id;    //下料命令序号
} sMsgData
(4) 应答帧命令：0x9B08
(5) 数据格式定义
typedef struct{
    unsigned char cmd[11];
    unsigned int32 package_id; (包裹号,板卡收到 ACK 或者达到最大重发次数才会从队列
    中移除此包裹的分拣结果，需要 3 次最多重发功能配合使用)
} sPC2Board_unload_Result_ACK;
```

## 2.27 从板未执行主板发送动作

(1) 发送时间

主板卡检测到从板卡未能执行主板卡命令时上传上位机记录。

(2) 发送帧命令：0x1A09

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char board_id;
} sPC2Board_InvalidBoard;
```

| 变量类型          | 变量名      | 含义            |
|---------------|----------|---------------|
| unsigned char | cmd[11]  | 详见 1.1 通讯协议格式 |
| unsigned char | board_id | 未能执行主板命令的从板站号 |

(4) 应答帧命令：无需应答

### 2.28.1 分拣时是否加速功能初始化

(1) 发送时间

上位机需要设置时发送，若本次初始化上位机没有发送给板卡，则所有分拣口按照默认值工作。另外此功能可以为每个分拣格口单独设置，为了方便上位机配置，假设 6 个分拣格口（序号 1-6），上位机可以单独设置，2 号和 5 号格口启用分拣加速 1.5 倍功能，1 号 3 号 4 号 6 号格口使用默认值。格口号需要通过板卡 ID 和方向来定位。此参数需要写入 Flash 中。

(2) 发送帧命令：0x1A0B

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char sMsgData;
} sPC2Board_IsAccelerate;
```

| 变量类型          | 变量名      | 含义            |
|---------------|----------|---------------|
| unsigned char | cmd[11]  | 详见 1.1 通讯协议格式 |
| sMsgData      | msg_data | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char BoardID; （分拣板卡 ID）
    unsigned char dir; （分拣的左右方向）
    unsigned char SpeedUpType; (IO 加速对应 0 485 加速对应 1)
    unsigned char SpeedupRatio; （加速比,0x0A 为基准 1.0 倍速度,0x11 为 1.7 倍速度）（对于只能加速或者不能加速的摆轮，0x0A 为不加速，其余值均为加速）
} sMsgData
```

(4) 应答帧命令：0x9A0B

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_IsAccelerate_Ack;
```

## 2.28.2 分拣时是否加速功能配置读取

(1) 发送时间

上位机需要时主动发送。

(2) 发送帧命令：0x1A14

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char Board_id;
} sPC2Board_IsAccelerate_Upload;
```

| 变量类型          | 变量名      | 含义            |
|---------------|----------|---------------|
| unsigned char | cmd[11]  | 详见 1.1 通讯协议格式 |
| unsigned char | Board_id | 板卡 ID         |

(4) 应答帧命令：0x9A14

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char sMsgData;
} sPC2Board_IsAccelerate_Upload_ack;
```

| 变量类型          | 变量名      | 含义            |
|---------------|----------|---------------|
| unsigned char | cmd[11]  | 详见 1.1 通讯协议格式 |
| sMsgData      | msg_data | 消息数据部分        |

```
typedef struct{
    unsigned char BoardID; (分拣板卡 ID)
    unsigned char dir; (分拣的左右方向)
    unsigned char SpeedUpType; (IO 加速对应 0 485 加速对应 1)
    unsigned char SpeedupRatio; (加速比,0x0A 为基准 1.0 倍速度,0x11 为 1.7 倍速度)(对于只能加速或者不能加速的摆轮, 0x0A 为不加速, 其余值均为加速)
} sMsgData
```

## 2.30 备用光电分拣功能初始化

(1) 发送时间

上位机需要设置时发送, 若上位机没有发送给板卡, 则板卡按照默认值 (启用) 工作。另外此功能可以为每张从板卡单独设置, 为了方便上位机配置, 假设 5 张从板卡 (序号 2-6), 上位机可以单独设置, 5 号板卡进入备用光电分拣功能, 3 号 4 号 6 号板卡不启用此功能。

(2 号从板卡 无法用备用光电分拣)

(2) 发送帧命令: 0x1A0C

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    Unsigned char msg_num;
    unsigned char sMsgData[msg_num];
} sPC2Board_BackUp_EnableStatus;
```

| 变量类型          | 变量名               | 含义            |
|---------------|-------------------|---------------|
| unsigned char | cmd[11]           | 详见 1.1 通讯协议格式 |
| unsigned char | msg_num           | 消息个数          |
| sMsgData      | msg_data[msg_num] | 消息数据部分        |

消息部分数据格式如下:

```
typedef struct{
    unsigned char board_id; (对应的板卡站号 ID)
} sMsgData
```

(4) 应答帧命令: 0x9A0C

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
}SPC2Board_BackUp_EnableStatus_Ack;
```

2.31 驱动器站号组别设置

- (1) 发送时间  
上位机或者调试 demo 在有需要时发送，此参数需要掉电保存。
- (2) 发送帧命令：0x1A0D
- (3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID; （板卡站号）
    unsigned char GroupCount; （组别总数）
    unsigned char sMsgData[GroupCount];
} sPC2Board_GroupInfo;
```

| 变量类型          | 变量名                  | 含义            |
|---------------|----------------------|---------------|
| unsigned char | cmd[11]              | 详见 1.1 通讯协议格式 |
| unsigned char | BoardID              | 从站 ID 号       |
| unsigned char | GroupCount           | 组别总数          |
| sMsgData      | msg_data[GroupCount] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char Group_ID;
    unsigned char Driver_Num;
    unsigned char DriverID[Driver_Num];
} sMsgData
(4) 应答帧命令：0x9A0D
(5) 数据格式定义
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID; （板卡站号）
} sPC2Board_GroupInfo_Ack;
```

2.32 驱动器站号组别读取

- (1) 发送时间  
上位机或者调试 demo 在有需要时发送，方便现场人员调试。

(2) 发送帧命令：0x1A0E

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID;
} sPC2Board_GroupInfo_Get;
```

| 变量类型          | 变量名     | 含义            |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |
| unsigned char | BoardID | 板卡站号          |

(4) 应答帧命令：0x9A0E

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID; (板卡站号)
    unsigned char GroupCount; (组别总数)
    unsigned char sMsgData[GroupCount];
} sPC2Board_GroupInfo;
```

| 变量类型          | 变量名                  | 含义            |
|---------------|----------------------|---------------|
| unsigned char | cmd[11]              | 详见 1.1 通讯协议格式 |
| unsigned char | BoardID              | 从站 ID 号       |
| unsigned char | GroupCount           | 组别总数          |
| sMsgData      | msg_data[GroupCount] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char Group_ID;
    unsigned char Driver_Num;
    unsigned char DriverID[Driver_Num];
} sMsgData
```

2.33 上位机掉线强制异常口设置

(1) 发送时间

上位机有需要时发送，目的是防止现场没有开启上位机就开始供包，导致漏扫。**当收到的异常口数量为 0 时，代表取消强制异常掉线口。**

(2) 发送帧命令：0x1A0F

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char Port_Num;
```



```

        unsigned char PortID[Port_Num];
    } sPC2_SDS_ErrorSP;

```

| 变量类型          | 变量名              | 含义            |
|---------------|------------------|---------------|
| unsigned char | cmd[11]          | 详见 1.1 通讯协议格式 |
| unsigned char | Port_Num         | 异常口数量         |
| unsigned char | PortID[Port_Num] | 异常口具体 ID      |

(4) 应答帧命令: 0x9A0F

```

typedef struct{
    unsigned char cmd[11];
} sPC2_SDS_ErrorSP_ACK;

```

## 2.34 初始化信息丢失后的心跳

(1) 发送时间

板卡检测到自身丢失了初始化信号后每 5 秒发送一次, 发送此心跳时不需要发送正常心跳。

(2) 发送帧命令: 0x1151

(3) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
} sBoard2PC_Heart_Data;

```

| 变量类型          | 变量名     | 含义            |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |

(4) 应答帧: 无需应答

## 2.35 使用过程中初始化格口信息

(1) 发送时间

上位机距离上次初始化一定时间后 (防止多次初始化), 收到板卡自检丢失了初始化信息时发送。板卡收到此初始化消息时不需要进行初始化寻零操作。

(2) 发送帧命令: 0x1A11

(3) 数据格式定义

```

typedef struct{
    unsigned char cmd[11];
    unsigned char unloadPortTableNum;
    sUnloadPortPara unloadPortParaTable[unloadPortTableNum];
} sPC2Board_UnloadPortPara;

```

| 变量类型 | 变量名 | 含义 |
|------|-----|----|
|------|-----|----|

|                 |                                         |               |
|-----------------|-----------------------------------------|---------------|
| unsigned char   | cmd[11]                                 | 详见 1.1 通讯协议格式 |
| unsigned char   | unloadPortTableNum                      | 下料口个数         |
| sUnloadPortPara | unloadPortParaTable[unloadPortTableNum] | 具体的每个下料口的参数信息 |

每个下料口参数信息格式如下：

```
typedef struct{
    unsigned char unload_port_num; //下料口号
    unsigned char broad_num;      //从控站号
    unsigned char unload_port_dir; //下料口方向
} sUnloadPortPara;
(4) 应答帧命令：0x9A11
(5) 数据格式定义
typedef struct{
    unsigned char cmd[11];
} sPC2Board_UnloadPortPara_ACK;
```

| 变量类型          | 变量名     | 含义            |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |

## 2.36 是否启用急停按钮

(1) 发送时间

配置 Demo 主动发送给板卡. 急停按钮仅限于调节多驱电柜软急停时候使用, 代码默认使用 (1) 急停功能。

(2) 发送帧命令：0x1A10

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID;
    unsigned char Button_Exist;
    unsigned char Emergency_Button;
} sPC2Board_Emergency_Button_cmd
```

| 变量类型          | 变量名              | 含义                   |
|---------------|------------------|----------------------|
| unsigned char | cmd[11]          | 详见 1.1 通讯协议格式        |
| unsigned char | BoardID          | 板卡站号                 |
| unsigned char | Emergency_Button | 急停按钮值 (0: 不启用 1: 启用) |

(4) 应答帧命令：0x9A10

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char BoardID;
    unsigned char Emergency_Button;
```

```
} sPC2Board_Emergency_Button_cmd_ack;
```

| 变量类型          | 变量名     | 含义            |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |

### 3.0 不允许分拣命令

此协议：自动供包摆轮专用

(1) 发送时间

上位机收到相机信号发送给板卡部分包裹不允许分拣的信息

(2) 发送帧命令：0x1B05

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 time_delay;
    unsigned int16 forbid_unload_time;
} sPC2Board_forbid_unload_Data;
```

| 变量类型           | 变量名                | 含义                    |
|----------------|--------------------|-----------------------|
| unsigned char  | cmd[11]            | 详见 1.1 通讯协议格式         |
| unsigned int32 | msg_id             | 下料命令序号                |
| unsigned int16 | time_delay         | 从相机触发光电开始到收到下料信息经过的时间 |
| unsigned int16 | forbid_unload_time | 不允许分拣的持续时间            |

(4) 应答帧命令：0x9B05

(5) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_forbid_unload_Data_ACK;
```

| 变量类型          | 变量名     | 含义            |
|---------------|---------|---------------|
| unsigned char | cmd[11] | 详见 1.1 通讯协议格式 |

### 3.1 摆轮单机分拣结果回应

此协议：自动供包摆轮专用

(1) 发送时间

下位机执行下料动作后反馈给上位机

(2) 发送帧命令：0x1B06

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char msg_num;
    sMsgData msg_data[msg_num];
} sPC2Board_unload_Result;
```

| 变量类型          | 变量名               | 含义            |
|---------------|-------------------|---------------|
| unsigned char | cmd[11]           | 详见 1.1 通讯协议格式 |
| unsigned char | msg_num           | 回应的消息个数       |
| sMsgData      | msg_data[msg_num] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char cmd_type;    //回应消息的类型(0:正常分拣,1:叠件回流,2:不允许供包回流)
    Unsigned int16 unload_port_id;//分拣口号
} sMsgData
```

（4）应答帧命令：无需应答

3.2 供包台是否允许供件信息上传

此协议：自动供包摆轮专用

（1）发送时间  
初始化完成后主动上传一次所有供包台的状态，然后每当供包台允许供件时上传（小于 500 毫秒的状态改变不上传）

（2）发送帧命令：0x1A07

（3）数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned char msg_num;
    sMsgData msg_data[msg_num];
} sPC2Board_Supply;
```

| 变量类型          | 变量名               | 含义            |
|---------------|-------------------|---------------|
| unsigned char | cmd[11]           | 详见 1.1 通讯协议格式 |
| unsigned char | msg_num           | 消息数据个数        |
| sMsgData      | msg_data[msg_num] | 消息数据部分        |

消息部分数据格式如下：

```
typedef struct{
    unsigned char Feed_id;        //供包机 ID
    unsigned char AllowSupply;    //是否允许供包（0：不允许，1：允许）
} sMsgData
```

(4) 应答帧命令：无需应答

### 3.3 存储配置文件

此协议：自动供包摆轮共用

(1) 发送时间

上位机检测到配置文件需要更新并且板卡支持此功能时发送,收到后先清空旧 Flash。

(2) 发送帧命令：0x2A01

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_num;
    unsigned int32 msg_length;
} sPC2Board_Config;
```

(3) 数据包请求 (板卡按数据包顺序请求,从 0 到数据包个数-1)

帧命令: 0x2A02

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_id; (数据包 ID)
} sPC2Board_Config_Request;
```

(4) 数据包下发 (除了最后一个数据包数据区长度均为 1024 字节)

帧命令: 0x2A03

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_id; (数据包 ID)
    unsigned int16 msg_len; (数据区长度)
    unsigned char msg[msg_len];
} sPC2Board_Config_Send;
```

(6) 写入 Flash 中该数据包对应的区域

(6) 配置文件下发结束

帧命令: 0x2A04

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_num;
    unsigned int32 msg_length;
} sPC2Board_Config_Finished;
```

全部配置数据包下发完成后或者板卡请求数据包 2 秒后仍未下发时,结束本次配置文件下发。

PS: 没有完整下发配置文件就中断时,清除文件配置 Flash.

### 3.4 读取配置文件

此协议：自动供包摆轮共用

(1) 发送时间

上位机需要时主动发送。

(2) 发送帧命令

0x2A05

(3) 数据格式定义

```
typedef struct{
    unsigned char cmd[11];
} sPC2Board_Config_Request;
```

(4) 上传缓存数据包参数（延迟 50 毫秒后再上传单个数据包）

帧命令：0x2A06

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_num;
    unsigned int32 msg_length;
} sPC2Board_Config_Info;
```

(5) 上传数据包（每次上传间隔 50 毫秒,按顺序传,除了最后一个数据包数据区长度均为 1024 字节）

帧命令：0x2A07

```
typedef struct{
    unsigned char cmd[11];
    unsigned int16 msg_id;（数据包 ID）
    unsigned int16 msg_len;（数据区长度）
    unsigned char msg[msg_len];
} sPC2Board_Config_upload;
```

# \*关于重发机制的说明

上位机和下位机重发间隔时间均为 300 毫秒一次

下位机重新发送时

第一次发送帧序号为 0x00 0x00 0x00 0x00

第二次发送帧序号为 0x01 0x00 0x00 0x00

第三次发送帧序号为 0x02 0x00 0x00 0x00

上位机重发时帧序号不变，但是需要重新计算包裹延迟时间

|         |                                                                                                     |
|---------|-----------------------------------------------------------------------------------------------------|
| 版本      | 更新说明                                                                                                |
| V1.0.04 | 增加到靠边区时间（3.3 节）                                                                                     |
| V1.0.05 | 增加关闭下料口功能（3.4 节）<br>变更分拣结果回应协议（2.5 节）                                                               |
| V1.0.06 | 细化报警信息（2.6.1 节,2.6.2 节）                                                                             |
| V1.0.07 | 新增可设置 BOOT 参数（3.1 节,3.2 节）                                                                          |
| V1.0.08 | 需改增加 2. 6. 3~2. 6. 4 节无 0 带驱动器报警相关                                                                  |
| V1.0.09 | 新增 2. 4 分拣命令 photo_trigger_time 字段                                                                  |
| V1.0.10 | 新增 3. 5 双驱动器调速命令                                                                                    |
| V1.0.11 | 新增 3. 6 双驱动器获取速度命令<br>新增 3. 7 摆轮线整体运行状态查询<br>BOOT 参数无需进入测试模式即可读写                                    |
| V1.0.12 | 新增 3. 8 线体启停控制, 新增分拣结果: 下料失败                                                                        |
| V1.0.13 | 新增 2. 6. 4 摆轮报警 2, 新增 3. 6 驱动器速度查询<br>3. 7 启停查询在状态改变时主动上报上位机                                        |
| V1.0.14 | 新增 2. 2. 1 初始化摆轮信息; 2. 6. 4 新增接近开关报警; 3. 6 驱动器速度反馈 具体到每一台摆轮的驱动器速度; 3. 1、 3. 2 读写摆轮参数具体到读写具体某一台摆轮的参数 |
| V1.0.15 | 新增 3. 9 摆轮急停状态查询                                                                                    |
| V1.0.16 | 新增 2. 2. 2 最小间隔设置命令, 新增分拣结果: 间隔过小                                                                   |
| V1.0.17 | 新增分拣结果: 从板执行分拣动作超时                                                                                  |
| V1.0.18 | 新增间距过小的包裹检测结果输出                                                                                     |
| V1.0.19 | 新增包裹长度（遮挡时间）输出                                                                                      |
| V1.0.20 | 新增板卡控制模式信息查询上传                                                                                      |
| V1.0.21 | 新增包裹补格口数据上传上位机                                                                                      |
| V1.0.22 | 新增不允许分拣命令和单机分拣结果命令                                                                                  |
| V1.0.23 | 1. 新增板卡版本号输出<br>2. 新增上位机设置光电滤波时间功能<br>3. 新增供包台是否允许供件信息上传功能                                          |

|         |                                                                                                                                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <ul style="list-style-type: none"> <li>4. 新增超时包裹数据追加上传功能</li> <li>5. 包裹间距过小检测增加开关功能</li> <li>6. 包裹分拣命令和分拣结果上传优化 ACK 方法</li> <li>7. 新增从板未执行主板发送动作功能</li> <li>8. 新增备用光电分拣结果上传功能</li> <li>9. 新增光电信息初始化功能</li> </ul> |
| V1.0.24 | <ul style="list-style-type: none"> <li>1. 新增从板是否备用光电分拣功能</li> <li>2. 新增格口是否加速分拣功能</li> </ul>                                                                                                                     |
| V1.0.25 | <ul style="list-style-type: none"> <li>1. 分拣时加速功能调整</li> <li>2. 可以设置和读取板卡内部的驱动器组别和站号</li> <li>3. 可以设置上位机掉线异常口</li> </ul>                                                                                         |
| V1.0.26 | <ul style="list-style-type: none"> <li>1. 重新定义了上传和设置板卡 boot 参数</li> </ul>                                                                                                                                        |
| V1.0.27 | <ul style="list-style-type: none"> <li>1. 新增板卡自检异常心跳</li> <li>2. 新增板卡丢失初始化信息重新初始化功能</li> <li>3. 掉线异常口配置改动</li> </ul>                                                                                             |
| V1.0.28 | <ul style="list-style-type: none"> <li>新增双驱是否加速功能</li> </ul>                                                                                                                                                     |
| V1.0.29 | <ul style="list-style-type: none"> <li>1. 新增左右摆可以按照设定脉冲摆动</li> <li>2. 新增机械间隙补偿功能</li> <li>3. 新增摆轮站号代码可选择功能</li> </ul>                                                                                            |
| V1.0.30 | <ul style="list-style-type: none"> <li>1. 新增急停按钮是否启用配置</li> </ul>                                                                                                                                                |
| V2.0.1  | <ul style="list-style-type: none"> <li>1. 删除 2.25 光电档位配置</li> <li>2. 左右角度配置优化</li> <li>3. 是否加速功能优化</li> <li>4. 是否加速配置可读取</li> <li>5. 左右角度配置可读取</li> </ul>                                                        |
| V2.0.2  | <ul style="list-style-type: none"> <li>1. 新增摆轮单机分拣结果类型</li> </ul>                                                                                                                                                |
| V2.0.3  | <ul style="list-style-type: none"> <li>1. 新增配置文件存入/读取功能</li> </ul>                                                                                                                                               |